CLAIMS

What is claimed is:

1. A method, comprising:

inserting an allocation instruction within a routine if a function call instruction is found within said routine.

2. The method of claim 1 further comprising configuring said allocation instruction to allocate only for the live information that exists within said routine when said inserted allocation instruction is executed.

3. The method of claim 2 wherein said live information is determined by identifying information that is referred to before and after said function call.

4. The method of claim 3 wherein said information identified after said function call extends to an exit block of said routine.

5. The method of claim 4 wherein the worst case path to said exit block is allocated for.

6. The method of claim 3 wherein said information identified after said function call extends to a post-dominator block of said routine.

7. The method of claim 6 wherein the worst case path to said post-dominator block is allocated for.

0960343-090300

8. The method of claim 2 wherein said live information is information that is local to said routine.

9. The method of claim 8 wherein the processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

10. The method of claim 2 wherein said live information includes global information.

11. The method of claim 1 wherein said allocation instruction is inserted just before said function call.

12. The method of claim 1 wherein said allocation instruction is inserted in a pre-dominator basic block of said function call.

13. The method of claim 12 wherein said allocation instruction is inserted in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

14. A method comprising:

inserting multiple allocation instructions within a routine by searching for one or more functional characteristics within said routine and inserting an allocation instruction within said routine if a functional characteristic is discovered.

15. The method of claim 14 wherein a loop in a control flow graph corresponds to a functional characteristic.

16. The method of claim 15 wherein said allocation instruction inserted for said loop is inserted above said loop.

17. The method of claim 16 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

18. The method of claim 16 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

19. The method of claim 14 wherein a software pipelined loop corresponds to a functional characteristic.

20. The method of claim 19 wherein said allocation instruction inserted for said software pipelined loop is inserted above said loop.

21. The method of claim 20 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

22. The method of claim 21 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

23. The method of claim 14 wherein a function call corresponds to a functional characteristic.

24. The method of claim 14 further comprising determining the number of registers to be allocated for an allocation instruction after a functional characteristic is found.

25. The method of claim 24 wherein all functional characteristics within said routine are discovered before said determining is performed.

26. The method of claim 24 wherein said determining is performed before a next functional characteristic is discovered.

27. The method of claim 14 further comprising building an understanding of said routine's control flow graph before said searching is performed.

28. A method, comprising:

a) performing a first allocation for a first amount of register space at the entry block of a routine;

b) performing a second allocation for a second amount of register space for the live information within said routine at the time of said second allocation;

c) performing a function call to a second routine;

d) performing a third allocation for a third amount of register space at the entry block of said second routine, said third amount of register space and said first amount register space having a common register.

29. The method of claim 28 wherein said live information is determined by identifying information that is referred to before and after said function call.

30. The method of claim 29 wherein said information identified after said function call extends to an exit block of said routine.

31. The method of claim 30 wherein the worst case path to said exit block is allocated for.

32. The method of claim 29 wherein said information identified after said function call extends to a post-dominator block of said routine.

33. The method of claim 32 wherein the worst case path to said post-dominator block is allocated for.

34. The method of claim 28 wherein said live information is information that is local to said routine.

35. The method of claim 34 wherein the processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global

information, said allocation instruction pertaining only to said register space used for local information.

36. The method of claim 28 wherein said live information includes global information.

37. The method of claim 28 wherein said second allocation is performed just before said function call.

38. The method of claim 28 wherein said second allocation is performed in a pre-dominator basic block of said function call.

39. The method of claim 28 wherein said second allocation is performed in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

40. The method of claim 28 further comprising compiling said routine.

41. A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:

inserting an allocation instruction within a routine if a function call instruction is found within said routine.

42. The machine readable medium of claim 41 further comprising sequences of instructions which cause the system to configure said allocation instruction to allocate only for the live information that exists within said routine when said inserted allocation instruction is executed.

43. The machine readable medium of claim 42 wherein said live information is determined by identifying information that is referred to before and after said function call.

44. The machine readable medium of claim 43 wherein said information identified after said function call extends to an exit block of said routine.

45. The machine readable medium of claim 44 wherein the worst case path to said exit block is allocated for.

46. The machine readable medium of claim 43 wherein said information identified after said function call extends to a post-dominator block of said routine.

47. The machine readable medium of claim 46 wherein the worst case path to said post-dominator block is allocated for.

48. The machine readable medium of claim 42 wherein said live information is information that is local to said routine.

49. The machine readable medium of claim 48 wherein the processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

50. The machine readable medium of claim 42 wherein said live information includes global information.

51. The machine readable medium of claim 41 wherein said allocation instruction is inserted just before said function call.

52. The machine readable medium of claim 41 wherein said allocation instruction is inserted in a pre-dominator basic block of said function call.

53. The machine readable medium of claim 52 wherein said allocation instruction is inserted in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

54. A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:

inserting multiple allocation instructions within a routine by searching for one or more functional characteristics within said routine and inserting an

allocation instruction within said routine if a functional characteristic is discovered.

55. The machine readable medium of claim 54 wherein a loop in a control flow graph corresponds to a functional characteristic.

56. The machine readable medium of claim 55 wherein said allocation instruction inserted for said loop is inserted above said loop.

57. The machine readable medium of claim 56 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

58. The machine readable medium of claim 56 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

59. The machine readable medium of claim 54 wherein a software pipelined loop corresponds to a functional characteristic.

60. The machine readable medium of claim 59 wherein said allocation instruction inserted for said software pipelined loop is inserted above said loop.

61. The machine readable medium of claim 60 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

62. The machine readable medium of claim 61 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

63. The machine readable medium of claim 54 wherein a function call corresponds to a functional characteristic.

64. The machine readable medium of claim 54 further comprising sequences of instructions which cause the system to determine the number of registers to be allocated for an allocation instruction after a functional characteristic is found.

65. The machine readable medium of claim 64 wherein all functional characteristics within said routine are discovered before said determining is performed.

66. The machine readable medium of claim 64 wherein said determining is performed before a next functional characteristic is discovered.

67. The machine readable medium of claim 54 further comprising sequences of instructions which cause the system to build an understanding of said routine's control flow graph before said searching is performed.